Chapter 5 C++ Functions and Header Files

Contents

Time required: 60 minutes

Online Tutorials

Go through the following tutorials before going through the tutorial assignments.

- C++ Functions
- C++ Function Parameters
- C++ Function Overloading

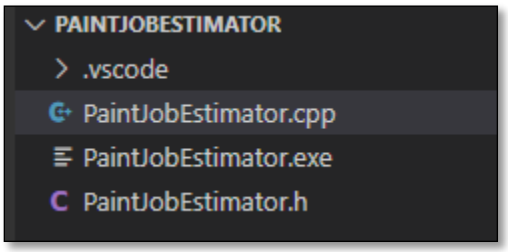
Compile Multiple CPP Files in VSCode

There is a zipped up .vscode file attached to this assignment. Unzip the .vscode file and place the folder in your project. This tells VSCode how to compile multiple files in CPP.

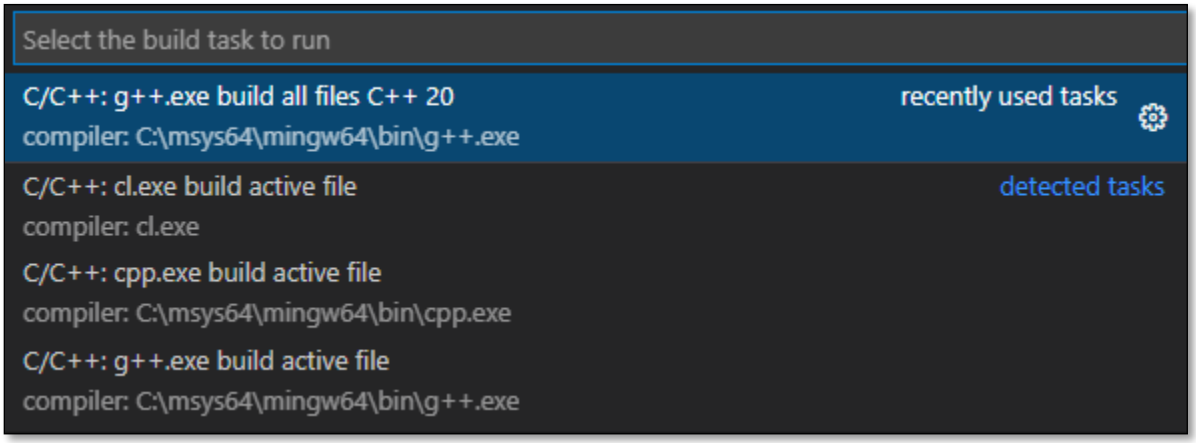
- Always open the folder that contains your program files.

- Create separate folders for each of your different languages.
- Create a separate folder for any CPP project that has multiple files.
- Copy the .vscode folder into any CPP projects that have separate files.

Your project will look something like this.



1. Open the project folder.
2. Open the cpp file.
3. **Terminal → Run Build Task**



4. Choose the highlighted option → **g++.exe build all files C++ 20**

Functions

- Functions are building blocks
- Also called modules, methods, procedures, or sub-procedures
- Like miniature programs
- Can be put together to form larger program
- One of two principle means of modularization in C++ (other is classes)

Why Use Functions?

Divide and Conquer

- Allow complicated programs to be divided into manageable components
- Programmer can focus on just the function: develop it, debug it, and test it
- Various developers can work on different functions simultaneously

Reusability

- Can be used in more than one place in a program--or in different programs
- Avoids repetition of code, thus simplifying code maintenance
- Can be called multiple times from anywhere in the program

Components:

- Custom functions and classes that you write
- Prepackaged functions and classes available in the C++ Standard Library

User Defined Functions

There can be 4 different types of user-defined functions, they are:

- Function with no arguments and no return value
- Function with no arguments and a return value
- Function with arguments and no return value
- Function with arguments and a return value

Tutorial 1: Overloaded Functions

Create, compile, run and attach the following program.

```cpp
/**
* Filename: PrintDecorator.cpp
* Written by:
* Written on:
* Purpose: Function headers and overloaded functions
*
*/

#include <iostream>
using namespace std;

// Overloaded function prototypes
void printline();
void printline(int len);
void printline(int len, string s);

int main()
{
    printline();
    printline(30);
    printline(20, "===");
    return 0;
}

void printline()
{
    cout << "+";
    for (int i = 0; i < 30; i++)
    {
        cout << "--";
    }
    cout << "+" << endl;
}

void printline(int len)
{
    cout << "+";
    for (int i = 0; i < len; i++)
    {
        cout << "**";
    }
    cout << "+" << endl;
}

```

```
45 void printline(int len, string s)
46 {
47     cout << "+";
48     for (int i = 0; i < len; i++)
49     {
50         cout << s;
51     }
52     cout << "+" << endl;
53 }
```

Tutorial 2: Math to C++ - Agile Development

How do you solve a problem in programming? Small step by small step by small step. We are going to develop a math program in incremental pieces. This is a good development process to use. Solve one thing at a time, then move to the next. The process shown is also called refactoring.

NOTE: This type of development process is known as AGILE development.

Let's see how we can convert some math into C++ programming. Let's start with calculating an exponent table.

If n is a positive integer and x is any real number, then x^n corresponds to repeated multiplication:

$$x^n = x \times x \ldots x \times x$$

We can call this "x raised to the power of n," "x to the power of n," or simply "x to the n." Here, x is the base and n is the exponent or the power.

How do we convert $x^n = x \times x \ldots x \times x$ into a computer program? Step by step.

1. x * x is x^2, x * x * x is x^3, etc. Each time we add another exponent (multiply the number one more times itself), we add another base into the calculation.
2. In programming, we have libraries to do some of the heavy lifting for us. The C++ cmath pow function for example.
3. How do we do repeated things in programming? Loops.

Pseudocode:

```
# Exponent table of 2 to the power of 10
base = 2
for i = 0-10
```

```
    num = pow(base, i)
    print(num)
```

1. Simple program with no input

This is the first step. Hard code everything, just get the basic program to work.

```
/**
* Filename: ExponentTable1.cpp
* Written by:
* Written on:
* Revised:
* Calculate exponent table
*/

#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    int base = 2;
    for (int i = 0; i < 10; i++)
    {
        int num = 0;
        num = pow(base, i);
        cout << base << " to the power of " << i << " is: " << num << endl;
    }
    return 0;
}
```

```
2 to the power of 0 is: 1
2 to the power of 1 is: 2
2 to the power of 2 is: 4
2 to the power of 3 is: 8
2 to the power of 4 is: 16
2 to the power of 5 is: 32
2 to the power of 6 is: 64
2 to the power of 7 is: 128
2 to the power of 8 is: 256
2 to the power of 9 is: 512
```

2. Program with user input

Add user input to the previous program.

```cpp
/**
* Filename: ExponentTable2UserInput.cpp
* Written by:
* Written on:
* Revised:
* Calculate exponent table
*/

#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    int base = 0;
    cout << "The Amazing Exponent Table Creator!" << endl;
    cout << "Please enter a whole number: ";
    cin >> base;
    for (int i = 0; i < 10; i++)
    {
        int num = 0;
        num = pow(base, i);
        cout << base << " to the power of " << i << " is: " << num << endl;
    }
    return 0;
```

```
The Amazing Exponent Table Creator!
Please enter a whole number: 2
2 to the power of 0 is: 1
2 to the power of 1 is: 2
2 to the power of 2 is: 4
2 to the power of 3 is: 8
2 to the power of 4 is: 16
2 to the power of 5 is: 32
2 to the power of 6 is: 64
2 to the power of 7 is: 128
2 to the power of 8 is: 256
2 to the power of 9 is: 512
```

3. Add a menu

Add a menu to allow the user to run the program more than once.

```cpp
/**
* Filename: ExponentTable3Menu.cpp
* Written by:
* Written on:
* Revised:
* Calculate Exponent table from base number
*/

#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    int base = 0;
    cout << "The Amazing Exponent Table Creator!" << endl;
    cout << "Please enter a whole number: ";
    cin >> base;
    while ((base > 0))
    {
        for (int i = 0; i < 10; i++)
        {
            int num = 0;
            num = pow(base, i);
            cout << base << " to the power of " << i << " is: " << num << endl;
        }
        cout << "The Amazing Exponent Table Creator!" << endl;
        cout << "Please enter a whole number: ";
        cin >> base;
    }
    return 0;
}
```

```
The Amazing Exponent Table Creator!
Please enter a whole number: 4
4 to the power of 0 is: 1
4 to the power of 1 is: 4
4 to the power of 2 is: 16
4 to the power of 3 is: 64
4 to the power of 4 is: 256
4 to the power of 5 is: 1024
4 to the power of 6 is: 4096
4 to the power of 7 is: 16384
4 to the power of 8 is: 65536
4 to the power of 9 is: 262144
The Amazing Exponent Table Creator!
Please enter a whole number: 0
```

4. Functions

Divide your code into functions.

```cpp
/**
* Filename: ExponentTable4Functions.cpp
* Written by:
* Written on:
* Revised:
* Calculate exponent table
*/

#include <iostream>
#include <cmath>
using namespace std;

// Function prototypes
int getInput();
void displayTable(int base, int i, double num);

int main()
{
    int base = 0;
    base = getInput();

    while ((base > 0))
    {
        for (int i = 0; i < 10; i++)
        {
            int num = 0;
            num = pow(base, i);
            displayTable(base, i, num);
        }
        cout << "The Amazing Exponent Table Creator!" << endl;
        cout << "Please enter a whole number: ";
        cin >> base;
    }
    return 0;
}
```

```
37 int getInput()
38 {
39     int base = 0;
40     cout << "The Amazing Exponent Table Creator!" << endl;
41     cout << "Please enter a whole number: ";
42     cin >> base;
43     return base;
44 }
45
46 void displayTable(int base, int i, double num)
47 {
48     cout << base << " to the power of " << i << " is: " << num << endl;
49 }
```

5. Function Header File

Move your functions into a header file.

```cpp
/**
* Filename: ExponentTable.cpp
* Written by:
* Written on:
* Revised:
* Calculate exponent table
*/

#include <iostream>
#include <cmath>
// Include header file with functions
#include "ExponentTable.h"
using namespace std;

int main()
{
    int base = 0;
    base = getInput();

    while ((base > 0))
    {
        for (int i = 0; i < 10; i++)
        {
            int num = 0;
            num = pow(base, i);
            displayTable(base, i, num);
        }
        cout << "The Amazing Exponent Table Creator!" << endl
        cout << "Please enter a whole number: ";
        cin >> base;
    }
    return 0;
}
```

```
/**
* Filename: ExponentTable.h
* Written by:
* Written on:
* Revised:
* Calculate exponent program header file
*/

#ifndef EXPONENTABLE_H
#define EXPONENTABLE_H

#include <iostream>
using namespace std;

//=================== FUNCTION PROTYPES ====================================//
int getInput();
void displayTable(int base, int i, double num);

//======================== GET INPUT ======================================//
int getInput()
{
    int base = 0;
    cout << "The Amazing Exponent Table Creator!" << endl;
    cout << "Please enter a whole number: ";
    cin >> base;
    return base;
}

//======================== DISPLAY TABLE ==================================//
void displayTable(int base, int i, double num)
{
    cout << base << " to the power of " << i << " is: " << num << endl;
}

#endif // EXPONENTABLE_H
```

Success! You have completed the Agile development process on this program. Use this development process on future projects, you will be glad you did!

Assignment Submission

Submit all C++ code files and a screenshot of each program showing that they work.